How to use Register Protection on MPC5748G

Following modules have register protection mechanism:

Topic	Related module
System memory map	All
SIUL registers	SIUL
PMC Digital Interface (PMCDIG) registers	PMCDIG
Mode Entry (MC_ME) module registers	MC_ME
PLL Digital Interface (PLLDIG) registers	PLLDIG
Clock Monitor Unit (CMU) registers	CMU
Fast Oscillator Digital Interface (FXOSC) registers	FXOSC
Slow External Oscillator (SXOSC) registers	SXOSC
Memory Error Management Unit (MEMU_0 and MEMU_1) registers	MEMU
Clock Generation Module (MC_CGM) registers	MC_CGM
Reset Generation Module (MC_RGM) registers	MC_RGM
LPU controller (LPU_CTL) registers	LPU_CTL

Not all registers in a module can be protected. There are tables in reference manual describing which registers can be protected. This is example for MC_ME registers:

Table 79-5. Protected MC_ME registers

Register	Register size (bits)	Offset from module base address	Protected size (bits)
ME_MCTL	32	4h	32
ME_ME	32	8h	32
ME_IM	32	10h	32
ME_SAFE_MC	32	28h	32
ME_DRUN_MC	32	2Ch	32
ME_RUN0_MC	32	30h	32
ME_RUN1_MC	32	34h	32
ME_RUN2_MC	32	38h	32
ME_RUN3_MC	32	3Ch	32
ME_RUN_PC0-7	32	80h-9Ch	32
ME_PCTL0-105.	8	C0h–129h[1]	8
ME_CCTL1-3	16	1C6h-1CAh[1]	16

Each module that have register protection feature implemented has the following memory map:

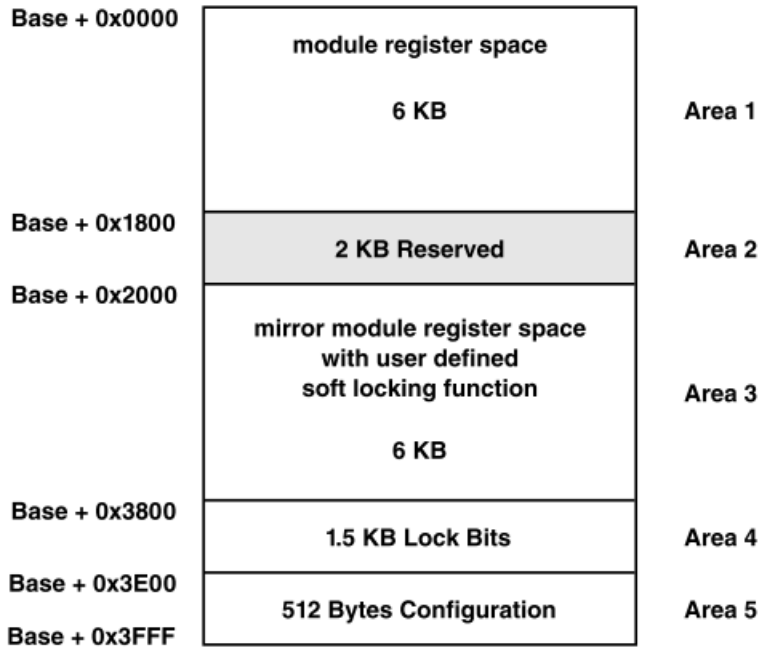


Figure 79-2. REG_PROT Memory Diagram

For example, we can find in reference manual that MC_ME module has base address 0xFFFB8000. That means the mirror module register space (Area 3 shown in the figure above) is at address 0xFFFB8000 + 0x2000 = 0xFFFBA000. Lock bits (Area 4) are at address 0xFFFB8000 + 0x3800 = 0xFFFBB800. The same applies to other modules.

Example:

Let's say we want to lock and unlock ME_RUN3_MC register.

Base address of MC_ME module is 0xFFFB8000.
Offset of ME_RUN3_MC register from base address is 0x3C.
ME_RUN3_MC register is at address 0xFFFB803C (base address + offset).
Mirrored address with soft locking function is at address 0xFFFBA03C (base address + 0x2000 + offset).
Lock bits for this register can be found at address 0xFFFBB80F.

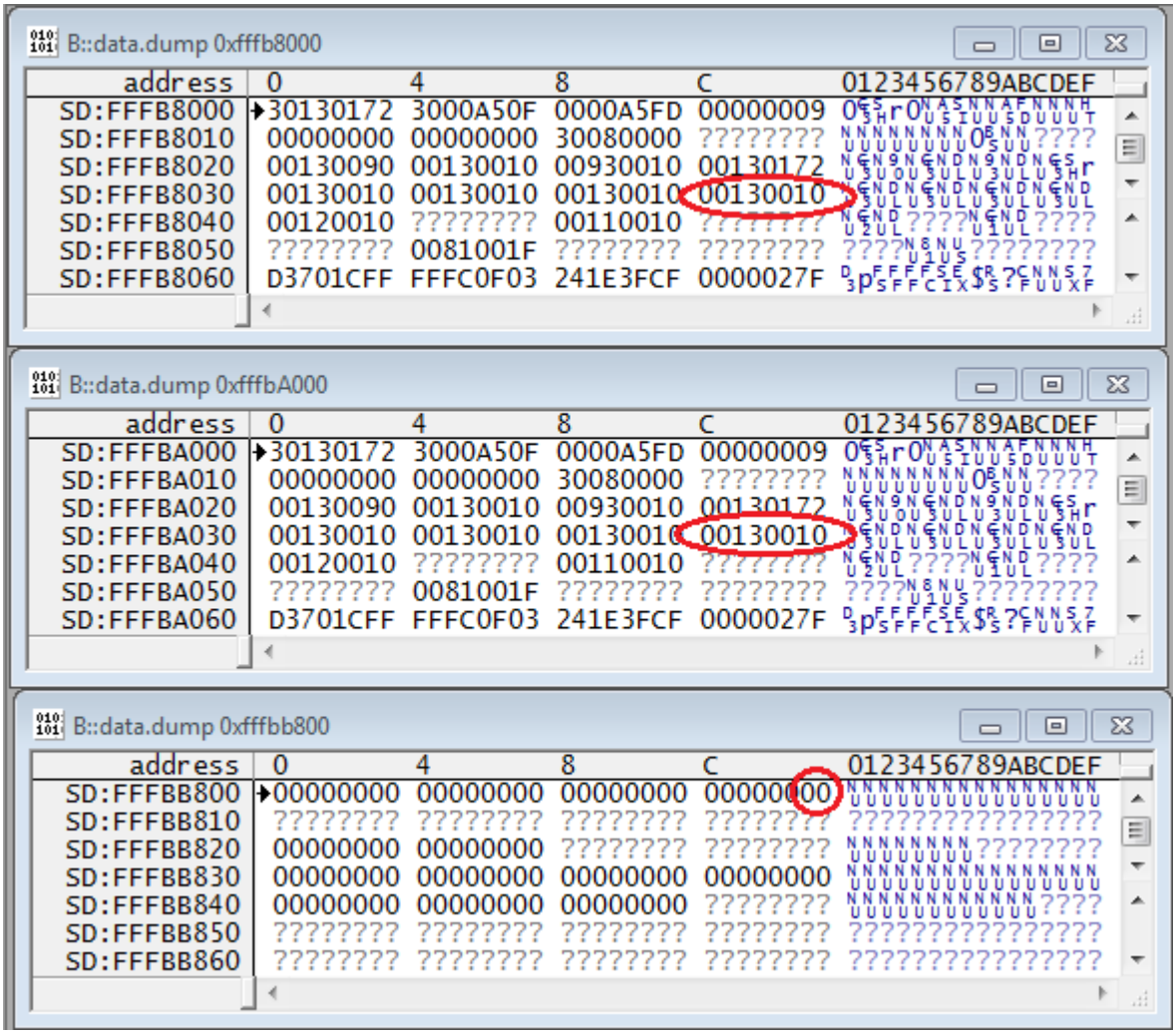
How to calculate address of lock register: each byte in Area 4 covers four bytes of module registers. See the description of REG_PROT_SLBRn register below. That means the ME_RUN3_MC register (32bit word) is covered by lock register at address: base address + 0x3800 + offset/4 = 0xFFFBB80F.

Bit	0	1	2	3	4	5	6	7
Read	0	0	0	0	SLB0	SLB1	SLB2	SLB3
Write	WE0	WE1	WE2	WE3				
Reset	0	0	0	0	0	0	0	0

REG_PROT_SLBRn field descriptions

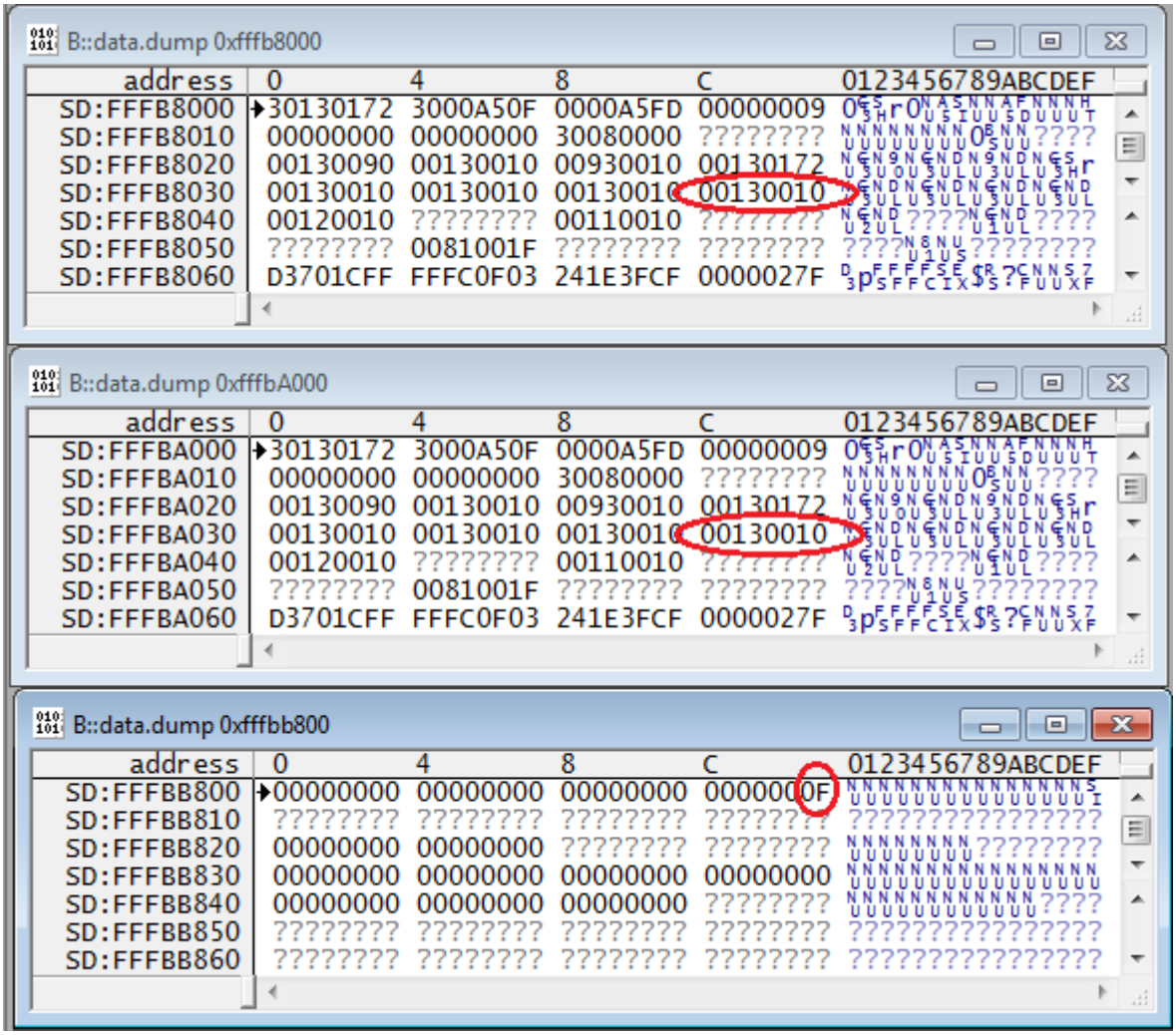
Field	Description
0 WE0	Write Enable Bits for Soft Lock Bits (SLB) WE0 enables writing to SLB0 1 Value is written to SLB 0 SLB is not modified
1 WE1	Write Enable Bits for Soft Lock Bits (SLB) WE1 enables writing to SLB1 1 Value is written to SLB 0 SLB is not modified
2 WE2	Write Enable Bits for Soft Lock Bits (SLB) WE2 enables writing to SLB2 1 Value is written to SLB 0 SLB is not modified
3 WE3	Write Enable Bits for Soft Lock Bits (SLB) WE3 enables writing to SLB3 1 Value is written to SLB 0 SLB is not modified
4 SLB0	Soft Lock Bits for one MRn register SLB0 can block accesses to MR[n *4 + 0] 1 Associated MRn byte is locked against write accesses 0 Associated MRn byte is unprotected and writable
5 SLB1	Soft Lock Bits for one MRn register SLB1 can block accesses to MR[n *4 + 1] 1 Associated MRn byte is locked against write accesses 0 Associated MRn byte is unprotected and writable
6 SLB2	Soft Lock Bits for one MRn register SLB2 can block accesses to MR[n *4 + 2] 1 Associated MRn byte is locked against write accesses 0 Associated MRn byte is unprotected and writable
7 SLB3	Soft Lock Bits for one MRn register SLB3 can block accesses to MR[n *4 + 3] 1 Associated MRn byte is locked against write accesses 0 Associated MRn byte is unprotected and writable

Below is a screenshot from debugger – we can see ME_RUN3_MC, its mirrored address and locking bits.



After reset, all lock bits are cleared, so protection is turned off. We can write to ME_RUN3_MC register as needed using the address 0xFFFB803C.
If we want to lock this register for further writes, there are two options:
First option is to write to mirrored address 0xFFFBA03C. After write, the register is automatically locked. The size of the register (protected size) is 32 bits, so all four bytes will be locked. SLB0, SLB1, SLB2 and SLB3 bits in REG_PROT_SLBR at address 0xFFFBB80F will be set.

When writing a register, it is good to respect the protected size. The protected size of ME_RUN3_MC register is 32 bits, so the width of write access should be 32bits. If the protected size is 8 bits, we should use byte access.



Second option is to write to REG_PROT_SLBR at address 0xFFFBB80F directly. To set SLB0-SLB3 bits in the register, we have to write value 0xFF to the register (we have to set WE0-3 bits to enable the write to SLBn bits).

In the same way we can clear the locking bits – we have to write value 0xF0 to the register REG_PROT_SLBR at address 0xFFFBB80F. This will unlock all four bytes in 32bit register ME_RUN3_MC.